Hybrid Data Management System for mHealth

Mervat Abu-Elkheir
Faculty of Computer and Information Sciences
Mansoura University
Mansoura, Egypt
mfahmy78@mans.edu.eg

Najah Abu Ali
Faculty of Information Technology
United Arab Emirates University
Al Ain, UAE
najah@uaeu.ac.ae

Karel Heurtefeux
Qatar Mobility Innovations Center
Doha, Qatar
karelh@qmic.com

Hamid Menouar
Qatar Mobility Innovations Center
Doha, Qatar
hamidm@qmic.com

***Abstract*—Mobile and wearable sensing technology stands to provide a wealth of information to healthcare providers, and allows them to envision systems with reduced costs, automated monitoring and evaluation, and overall improved healthcare services. However, the volume of data produced by such mobile and sensing technologies needs to be managed efficiently and continuously so as to realize its full potential in providing cutting-edge services. In this paper, we propose a mHealth data management system with the aim to provide near real-time operational an analytical services, while supporting long-term and offline processes and deep analytics. The components of the system will be discussed, and the potential workflows will be outlined.**

Keywords-component; data management; mHealth; IoT

I. Introduction

Mobile Health (or the more popular abbreviation, mHealth) refers to the use of mobile devices to support health services, such as self-monitoring [1] and activity recognition [2]. This involves collecting health data, delivering diagnostic and medical information to patients and healthcare providers, and providing remote patient monitoring. The accelerated market penetration of smartphones and tablets, as well as the evolution of more and more sophisticated and portable medical devices and sensors with integrated communication capabilities, is projected to trigger the generation of massive, diverse, and constantly streaming data volumes. The management of this data so as to provide the best possible health services becomes of paramount importance, due to the life-critical and inherently urgent nature of such services.

Data management refers to the architectures and procedures that are needed to properly manage the data lifecycle needs of a certain system. In the context of mHealth, data management should act as a layer between the medical devices and sensors generating health data and the applications and services accessing this data for operational as well as analytical purposes. Traditional data management systems handle the storage, retrieval, and update of elementary data items, records and files. In the context of mHealth, data management systems must summarize data online while providing storage, logging, and auditing facilities for offline analysis. This expands the concept of data management from offline storage, query processing, and transaction management operations into online-offline communication/storage dual operations.

mHealth, unlike traditional health information systems, has distinctive characteristics when it comes to the generated data that make traditional relational-based database management an inefficient solution. A massive volume of heterogeneous, streaming and geographically-dispersed real-time data will be created by millions of medical devices and sensors periodically sending observations about patients' vital signs or reporting the occurrence of potentially urgent medical events [3].

The lifecycle of data within an mHealth system proceeds from data production to aggregation, transfer, optional filtering and preprocessing, and finally to storage and archiving. Querying and analysis are the end points that initiate (request) and consume data, but data production can be set to be "pushed" or published to the mHealth consuming services [4]. Production, collection, aggregation, filtering, and some basic querying and preliminary processing functionalities are considered online, communication-intensive operations. Intensive preprocessing, long-term storage and archival, and in-depth processing/analysis are considered offline storage-intensive operations.

In this paper, we propose a mHealth data management system that provides near real-time operational an analytical services, while supporting long-term, offline processes and deep analytics. The proposed system is currently being implemented, with operational results and design challenges/lessons to be reported in future publications.

The remainder of the paper is organized as follows: in Section II, we discuss the characteristics of mHealth data and how they affect the design of the proposed system, and then the system itself is detailed. Section III illustrates example workflows of the proposed system, and Section IV concludes the paper.

MOBIHEALTH 2014, November 03-05, Athens, Greece

DOI 10.4108/icst.mobihealth.2014.257470

II. mHealth Data Management System

In a previous work [5], we proposed a number of design primitives that should be considered when developing a data management solution for IoT, and proceeded to propose a data management framework that takes these design primitives into consideration. In this work, we adapt the aforementioned data management framework for the needs of mHealth system, since mHealth systems can be considered one form of IoT.

In the following sections, we briefly discuss the characteristics of mHealth data and how they affect our design, and then proceed to illustrate the design of the proposed mHealth data management system.

A. Characteristics of mHealth Data

Health data is immutable; old values are not updated (i.e. replaced) with new ones. Rather, data is time-stamped and appended to the system, thereby creating a data stream. This data stream continuously captures the state of the entity generating the data, and acts as a recorded "history". Our system is built to incorporate this streaming nature of data, while still catering for offline, in-depth operations.

Data may be generated on the move. Patient movement is either restricted (e.g. inside a house) or extended beyond limited spaces. This dictates that we geo-tag the data, which is helpful for patient monitoring (as would be needed for Alzheimer's patients) and will also enable queries which are defined by geographical areas (e.g. for disease spread control).

Health data often needs to be aggregated/transformed in order to control its volume and filter any false or abnormal sensor readings. Only abnormal events of interest to healthcare providers are of essence. Therefore, data indicating normal conditions can be summarized with little effect on the accuracy of subsequent analysis, and therefore the real-time part of the system can be optimized. This "normal majority" also makes it relatively easy for the processing modules to pinpoint abnormalities and correct any outliers that are not statistically significant.

Health data can capture urgent and sudden onsets of health conditions as well as prolonged developments of health conditions. Therefore, two modes of operation need to be supported: offline to capture chronically developing conditions and patterns, and real-time to capture urgent incidents.

B. Functional Details

The proposed system relies on three main modules: an offline, batch processing module; an online, near real-time module; and a publish-subscribe module for data streams discovery and brokering. The components of the proposed system are illustrated in Fig. 1 and discussed below.

1) Smart and Wearable Medical Devices: Wearable sensors that can generate health monitoring data about vital signs, such as EEG, ECG, glucose levels, blood pressure. They can also generate data related to human activities, such as vision, hearing, positioning, and motor function. Smart implants can also provide valuable data about the health status of internal organs. Smart devices that are not necessarily wearable but still provide health-related data include health monitoring devices, smartphones, weight scales, and surveilence cameras. In our implementation, we start with ECG and activity monitoring, and will incorporate input from more sensors and devices as the system becomes operational.

2) Communication: Wearable devices will communicate their readings to gateways, such as smartphones or WiFi access points, via bluetooth or Zigbee technology. Gateways will then transmit medical readings to the primary care providers associated with patients/individuals, either via broadband or cellular 3G/4G connectivity. In our implementation of the system, the smartphone is the gateway of choice, due to its widespread penetration and ease of use. In addition, it is easily programmed to include applications that can manage other smart devices and sensors and support basic in-house analytics near the patient's physical perimeter.

3) Streams Publish-subscribe: The readings generated by wearable and smart devices will form streams of data. Each stream will be identified by its source device. At the same time, different sensory readings for the same patient at each time instance are kept as a record. This creates a patient stream that is composed of serialized records and the streams for different patients are uniquely identified by the patients' IDs and their permanent geographic locations (home address for example). The use of geographic locations as identifiers can be dynamic by refining their granularity to the level of individual patient records, which will reflect the real-time mobility of patients at the time the readings are generated. As patients acquire/install new sensors and smart devices, the system will publish the details of those devices to the Data Streams Publisher, which will add them to the Streams Repository. The system enables health applications and services to dynamically subscribe to the streams of specific sensors/devices across all patients, streams of specific patients, or streams generated at specific geographic areas, via the Data Streams Broker.

4) Storage: Each patient's data is serialized and stored as a separate stream. Data is identified by the timestamp, the device type generating the data, and the geographic location where the data is generated. Two types of data can be supported by the system: routine and periodic readings that are used for constant monitoring; and events signaling medical emergencies. We will support the storage of periodic sensor readings in the current implementation, and proceed to support medical events at a later stage, together with complex event processing capabilities. Data will be stored in a stream database, which will provide support for two modes of operation: an online, near real-time mode; and an offline, batch mode. In the online mode, data streams will be processed once as they are received by the system managed at the healthcare providers end. A queuing system will be used to manage the data window to be processed, and results are partial and dynamically updated as new data becomes available. In the offline mode, data will be processed as a whole, and results are computed once per query.

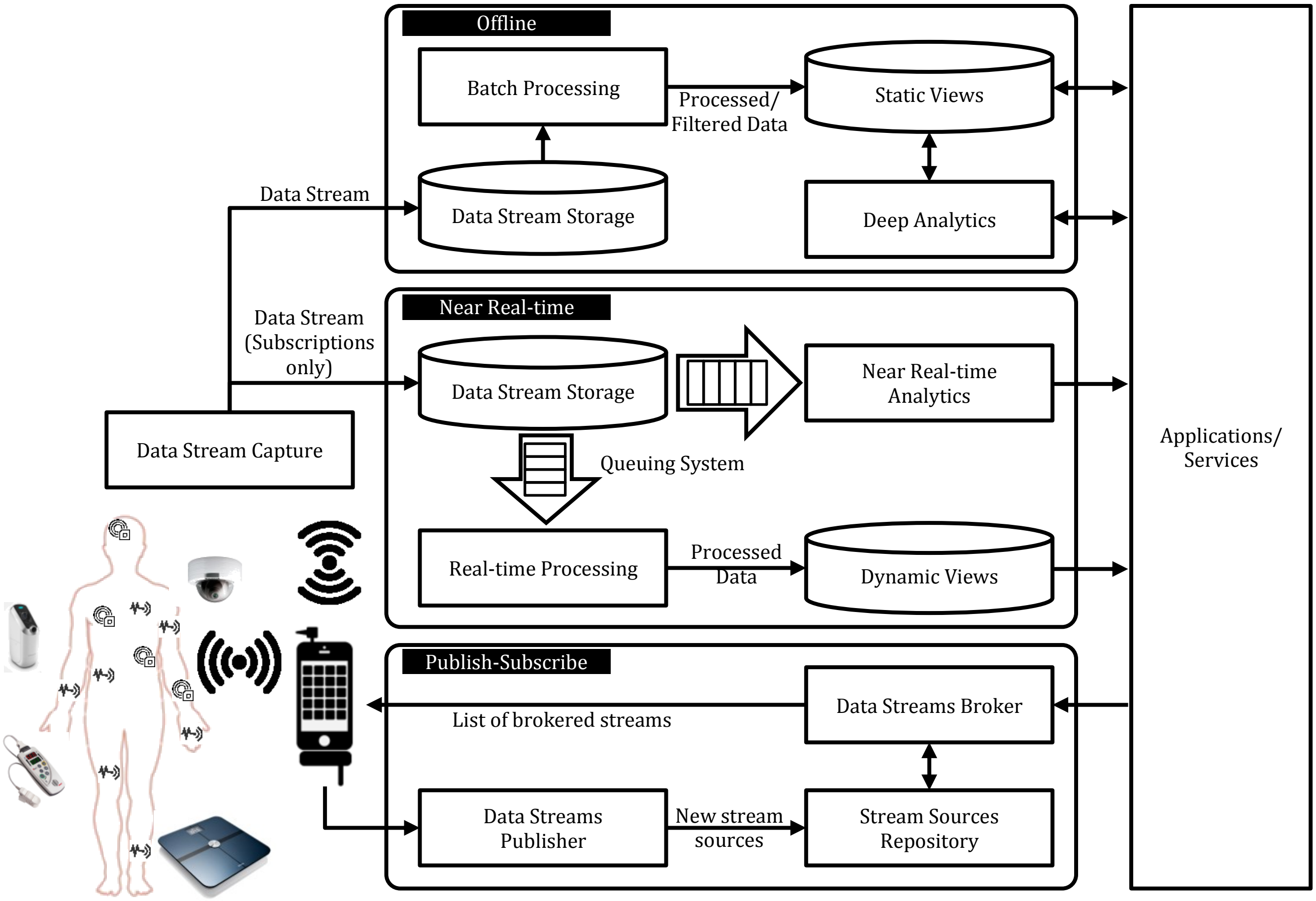

Figure 1. mHealth data management system.

5) Offline Processing Module: Within this module, the system can perform unlimited computations on the whole dataset in batches. This comes at the expense of high latency in producing meaningful results. Therefore, this module supports applications and services that execute sophisticated operational functions and perform deep analysis tasks that rely on complex algorithms such as the prediction of a chronic medical condition at its earliest stages or the estimation of the correlation between different vital signs based on the analysis of their corresponding streams. In order to support offline processing over the massive volume of medical data, we opt to use the following tools:

a) Hadoop Batch Processing: The Hadoop [1] system enables the distribution of processing tasks within a Hadoop cluster. The dataset is divided into subsets (e.g. groups of patient streams), and then a *Map* phase is performed, where an operational function is executed over all data subsets in parallel. This operational function can be as simple as finding the average reading for a specific sensor over all patients within the subset, but can extend to more complex processes as required by the applications and services. After the processing of the different subsets is complete, a *Reduce* phase is performed, in which the results from the different subsets are then combined into a single aggregated result. The Hadoop system can be used to efficiently perform preprocessing tasks that can be challenging to perform over big data, such as filtering abnormal or false readings, handling missing values, and performing data transformation tasks such as normalization or discretization.

b) Static View Database: The results of offline processing (e.g. summaries, filtered data, etc.) need to be stored in a read-only static view database, such as ElephantDB [2] or Cassandra [3]. Whenever batch processing workflows are rerun, the read-only views are not updated, but rather regenerated from scratch.

c) Deep Analytics: Operational processes do not provide enough insights into long-term health conditions; how they evolve and spread, and what may cause them. Therefore, sophisticated analysis algorithms need to be applied to the data in order to extract interesting correlations and previously unknown knowledge. Hadoop has been bolstered recently to support analytics via the Radoop[4] platform, which provides predictive machine learning and data mining processes in a Hadoop environment.

[1] http://hadoop.apache.org/

[2] https://github.com/nathanmarz/elephantdb

[3] http://cassandra.apache.org/

[4] http://www.radoop.eu/

6) Stream Processing Module: Stream processing involves continuous computations over the constantly generated data. For this purpose, only a limited window of data is stored and only for the duration of computation, which means that the real-time system should tolerate results produced from partitioned data. A queuing system such as Kafka[5] will be used to regulate the flow of data to the processes to be performed over the data, which does not guarantee a true real-time operation, but rather a near real-time one. To support stream processing, the following tools will be used in our system:

a) Storm Real-time Processing: Storm[6] is a real-time, stream processing system in Hadoop. Storm can efficiently handle multiple streams, or *Spouts*, and process those input streams in what is called *Bolts*, in order to produce output streams. These processes range from running functions to aggregation and filtering. System users define how to process the data through topologies (networks of spouts and bolts). The processing results can then be passed to Hadoop.

b) Dynamic View Database: The results of stream processing need to be stored in a read/write database that accepts new results as new data becomes available. Examples are Hbase[7] and Cassandra. We opt to use Cassandra, due to its support for both read-only and read-write modes.

c) Near real-time analytics: Analytics that are to be performed on data streams will not be as complex as deep analytics, and will involve basic aggregation, summarization, and filtering functionality. Incremental analytics, such as incremental clustering an association mining, can be performed in near real-time. Storm will provide good support for this level of functionality, and therefore will be the tool of choice. However, we are exploring the potential of building our own in-house online analytics solution with advanced functionality that can be merged with the results of deep analytics.

III. Example System Workflow

The wearable, implanted, or standalone health monitoring sensors and devices on the patient (who is considered an abstracted network of sensors), report on the patient's vital signs and possibly track her/his whereabouts as her location changes, either indoors or outdoors (e.g. while driving). These sensors are connected wirelessly to the patient's smartphone, which is considered a concentration point that captures and collects the data. Information analysts at the hospital use applications, services, and queries to run processes and analyze data for individual patients or collective patients' data. System users can set default stream subscriptions to work with in the near real-time mode, and change those subscriptions as needed.

Once subscriptions are set up by the Data Stream Broker, the smartphone proceeds to collect readings from the patient's sensors and medical devices. Vital readings that are collected periodically by the smartphone are reported wirelessly to the respective caregiver's network via a backbone network such as 3G/4G, and stored in the stream data store. Near real-time processing or analytics are then invoked to perform functions such as routine follow-up tasks or the discovery of interesting patterns related to possibly developing or spreading health conditions, such as post-op infection incidents related to operations performed at that given hospital. The outcomes and results of these processes are displayed on the caregivers' UI, and updated constantly as new data becomes available, or whenever the respective caregivers change the settings of the online system to include more/less data streams.

The offline module is used on demand, and only invoked whenever the system users need to run large-scale processes on the data. The results of these processes are stored in the static view, which is then used as input to complex analysis algorithms in the deep analytics component. Since the results of such analysis are not as volatile as is the case with online processing, they can be stored for long-term use, and the processes that generated them can be tweaked to provide more meaningful analysis.

IV. Conclusion

In this paper, we proposed a data management system for mHealth, and discussed its functional components and workflows. The proposed system supports offline and near real-time operations, as well as long-term analytics. The system is currently being implemented; with operational results to be reported in future publications. We plan to design a federated data management system that spans multiple, diverse, and geographically dispersed medical data stores. The purpose of this federated system will be to provide a more globalized view of medical data that will enable the detection of interesting city-wide, state-wide, country-wide, or even world-wide health patterns and conditions. This can serve to identify the existence of epidemics or seasonal symptoms, as well as visualize their spread rate, severity levels, and locale. This globalized analytics view will provide better support for disease surveillance, epidemic outbreak tracking, and the prompt containment of life-threatening conditions.

Acknowledgment

This work was made possible by NPRP grant #NPRP4-553-2-210 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] L.E. Burke et al., "Using mHealth Technology to Enhance Self-Monitoring for Weight Loss," *American Journal of Preventive Medicine*, vol. 43, no. 1, pp. 20–26, July 2012.

[2] Y. Hong, I. Kim, S.C. Ahn, and H. Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simulation Modelling Practice and Theory*, vol. 18, no. 4, pp. 446-455, April 2010.

[3] Geng Wu, S. Talwar, K. Johnsson, N. Himayat, and K.D. Johnson, "M2M: From mobile to embedded internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36-43, April 2011.

[4] IoT-A, "Converged architectural reference model for the IoT v2.0," European Commission Seventh Framework Programme, Project Deliverable 2012.

[5] M. Abu-Elkheir, M. Hayajneh, and N. Abu Ali, "Data Management for the Internet of Things: Design Primitives and Solution," *Sensors*, vol. 13, no. 11, pp. 15582-15612, November 2013.

[5] http://kafka.apache.org/

[6] http://hortonworks.com/hadoop/storm/

[7] http://hbase.apache.org/